

**COMP50315: SOFTWARE ENGINEERING FOR THE INTERNET 2018/19**

**Requirements Document**

**Project Name:** Durham University Sport (DUS)

**Team Number:** 7

**Team Members:**

Ce MA

Chang CHEN

Xinyi LI

Qing ZHANG

Zihan ZHAO

Table of Contents

[1. Summary 3](#_Toc8161118)

[1.1 Purpose 3](#_Toc8161119)

[1.2 Scope of Project 3](#_Toc8161120)

[1.3 Overview 3](#_Toc8161121)

[2. Domain Analysis 4](#_Toc8161122)

[2.1 Introduction 4](#_Toc8161123)

[2.2 General knowledge about the domain 4](#_Toc8161124)

[2.3 Customers and users 4](#_Toc8161125)

[2.4 The environment 4](#_Toc8161126)

[2.5 Tasks and procedures currently performed 4](#_Toc8161127)

[2.6 Competing website 5](#_Toc8161128)

[2.7 Similarities to other domains 5](#_Toc8161129)

[3. Solution Requirements 5](#_Toc8161130)

[3.1 Functional requirements 5](#_Toc8161131)

[3.2 Non-function requirements 10](#_Toc8161132)

[3.3 Functional Requirement Specification 11](#_Toc8161133)

[3.3.1 Guest Use Case 12](#_Toc8161134)

[3.3.2 User Use Case 13](#_Toc8161135)

[3.3.3 Administrator Use Case 15](#_Toc8161136)

[3.3.4 Trainer Use Case 17](#_Toc8161137)

[3.4 User Characteristics 17](#_Toc8161138)

[4. Data 18](#_Toc8161139)

[4.1 Data Description 18](#_Toc8161140)

[4.2 ERD 20](#_Toc8161141)

[5. Definition of terms 20](#_Toc8161142)

[6. Reference 21](#_Toc8161143)

# 1. Summary

## 1.1 Purpose

The purpose of this project is to present a detailed explanation of an online booking system. This project will try to foresee as many functional and non-functional requirements as possible to plan, organize and control functionalities so that the system is completed as successfully as possible in spite of all the risks.

## 1.2 Scope of Project

This software system is a web and mobile based application that is designed to meet the development needs of DUS by hiring out its facilities and holding events regularly. It allows users to view a calendar of events, get information and pricing on each facility, and make an online booking for these facilities.

Furthermore, only the authorized person, i.e. administrator is allowed to make changes on bookings and update the availability of facilities. The schedule will be available at the beginning of each academic year, when administrator and sports trainers can start block booking facilities for their courses.

The specific list of deliverables produced by the project will be further expressed below.

## 1.3 Overview

In this document, we shall first briefly introduce a domain analysis, this will be followed by a description of the target market and competitive environment. In the later section, the functional and non-functional requirements are discussed and then details on the user , administrator and trainer use cases are explored. Lastly, the tables for different database entities and database EDR diagram are precisely displayed.

# 2. Domain Analysis

## 2.1 Introduction

The domain is “Online Sports Facility Booking System”. This chapter describes the background information about this domain and the motivation for this analysis is to develop a new system that would improve existing systems.

## 2.2 General knowledge about the domain

1)   Most events will be held during working days.

2) Events or courses depending on the availability of facility (where and when an event or a course is to be held).

3) Setting a limit on the maximum number of participants that are permitted to be in an event or a course.

4) Updating the information of events or courses in real-time to reduce the risk of repeated postings or other clutters.

5) The privacy of stakeholders should be protected.

## 2.3 Customers and users

The main target clients of this website are Durham university students and sports tutors. Also, there are some potential customers that include citizens and students from other universities.

## 2.4 The environment

A majority of students and university staffs are either Windows or macOS users, thus the priority of our new booking system is to operate properly on both operating systems and be compatible with different browsers. Only a limit number of potential users tend to use other operating systems.

## 2.5 Current system

The official Durham sports website is operating with a manual system and has a team of staff member to serve their niche market. The website currently does not have an online booking system and the booking process can only take the form of phone calls, walk-in or emails. There is an inefficient system that all event bookings require customers to complete the same booking form. As a result, duplicate copies of the same information are created and hard to keep. Additionally, all the events or courses details shown on the website are unclear and unspecific to customers, and this massive of writing works might increase human error.

The functions currently performed by the manual system are:

* Facility Details.
* Booking for an event.
* Admission of New customer.
* Hall assigning related to user’s need.
* Customers’ personal profile.

## 2.6 Competing website

We have investigated several websites in this domain, the common functions of them are listed below:

1)  User registration and account management system.

2)  Booking events or public courses with inputting or selecting information of date, location (facility) and contact person (user or trainer).

3)  Booking records management system.

4)  Cancel/update events and courses.

5)  Searching items with specific details.

However, these websites are designed without functions like updating the facility capacity, visualizing an event calendar or sending an email verification, leading to problems such as an inefficient reservation and a certain degree of data redundancy.

## 2.7 Similarities to other domains

The pattern of this website is similar to the online shopping website and the online ticket system, both require user registration and records management.

# 3. Solution Requirements

## 3.1 Functional requirements

|  |  |
| --- | --- |
| **ID, type and title** | FR1 Website application - Sign up |
| **Description** | The guest can access the website to sign up, which allows him to have the same features as a client. |
| **Priority** | High |
| **Dependencies** | N/A |

|  |  |
| --- | --- |
| **ID, type and title** | FR2 Website application - Sign in |
| **Description** | A user can access the website and use his account details (ID, e-mail, password) to sign in. |
| **Priority** | High |
| **Dependencies** | FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR3 Website application - Update details |
| **Description** | The user can update personal details in personal profile. After signing in, a user should be able to go to his personal profile page where he can see his information and all the booking records. |
| **Priority** | High |
| **Dependencies** | FR1 and FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR4 Website application - Reset password |
| **Description** | The user can reset password. If a user forgets his password when signing in, the system should be able to send him an email which contains a link to reset password. |
| **Priority** | High |
| **Dependencies** | FR1 |

|  |  |
| --- | --- |
| **ID, type and title** | FR5 Website application - Search information |
| **Description** | The user/guest can search information. When inputting keywords in the search text area the user/guest should be able to see all related facilities or classes/sessions and appointments. |
| **Priority** | Mid |
| **Dependencies** | FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR6 Website application - Make online booking |
| **Description** | The user can make an online booking based on the availability of facility. The user should be able to book facilities one week in advance. |
| **Priority** | High |
| **Dependencies** | FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR7 Website application - Send confirmation email |
| **Description** | The user can receive emails when they finish online booking. The email should be able to include the confirmation of bookings and the total cost. |
| **Priority** | Mid |
| **Dependencies** | FR2 and FR6 |

|  |  |
| --- | --- |
| **ID, type and title** | FR8.1 Website application - View calendar |
| **Description** | The user can see all bookings on the calendar for the activities listed in 'other bookings' below. In addition, the user should have the ability to view calendar for the available facilities. |
| **Priority** | Mid |
| **Dependencies** | FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR8.2 Website application - View calendar periodically |
| **Description** | The user can see activities in the calendar via daily, weekly and monthly views. |
| **Priority** | High |
| **Dependencies** | FR2 and FR8 |

|  |  |
| --- | --- |
| **ID, type and title** | FR9 Website application - Failed book because maximum capacity |
| **Description** | The user cannot book the facility if the maximum capacity of it has been reached. |
| **Priority** | Mid |
| **Dependencies** | FR2 and FR6 |

|  |  |
| --- | --- |
| **ID, type and title** | FR10 Website application - Receive an apology email. |
| **Description** | The user should be able to receive an apology email if the booked facility is out of work. |
| **Priority** | Mid |
| **Dependencies** | FR2 and FR6 |

|  |  |
| --- | --- |
| **ID, type and title** | FR11.1 Website application - Add, edit, delete facilities |
| **Description** | Administrator can add, edit, delete facilities on the website. |
| **Priority** | High |
| **Dependencies** | FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR11.2 Website application - Update prices |
| **Description** | Administrator can update prices of the facilities on the website. |
| **Priority** | High |
| **Dependencies** | FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR11.3 Website application - Upload images |
| **Description** | Administrator can upload images of the facilities on the website. |
| **Priority** | High |
| **Dependencies** | FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR12.1 Website application - Add a booking (email) |
| **Description** | Administrator can add a booking that is taken by email. |
| **Priority** | High |
| **Dependencies** | FR2 and FR6 |

|  |  |
| --- | --- |
| **ID, type and title** | FR12.2 Website application - Add a booking (phone) |
| **Description** | Administrator can add a booking that is taken by phone. |
| **Priority** | High |
| **Dependencies** | FR2 and FR6 |

|  |  |
| --- | --- |
| **ID, type and title** | FR12.3 Website application - Cancel booking |
| **Description** | Administrator can cancel any booking manually. |
| **Priority** | High |
| **Dependencies** | FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR13 Website application - Block booking |
| **Description** | Administrator can block booking for the facilities and can also block booking of a facility in periods of time. |
| **Priority** | Mid |
| **Dependencies** | FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR14 Website application - Manage announcements |
| **Description** | Administrators can manage announcements including publish, delete, and update announcements. The administrator can sort the announcements. |
| **Priority** | Low |
| **Dependencies** | FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR15 Website application - View calendar |
| **Description** | Administrator can view one overall calendar for all facilities or toggle between calendars. Administrator can view the calendar for one or more specific facilities. |
| **Priority** | Mid |
| **Dependencies** | FR2 |

|  |  |
| --- | --- |
| **ID, type and title** | FR16.1 Website application - Book facilities (Trainer) |
| **Description** | Trainers can book a facility in periods of time and update details of it. The trainer must book facilities before the academic year commences. |
| **Priority** | Low |
| **Dependencies** | FR2 and FR6 |

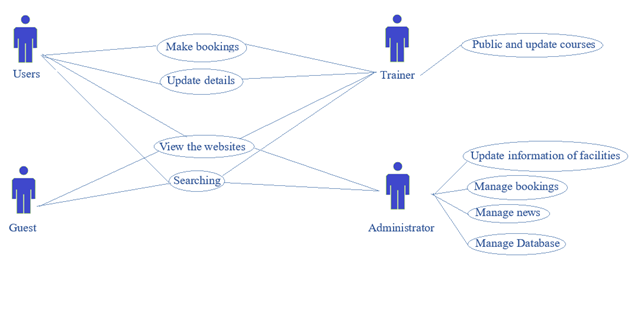
|  |  |
| --- | --- |
| **ID, type and title** | FR16.2 Website application - Publish courses (Trainer) |
| **Description** | The trainer can publish and update the latest status of the course, including the introduction of the course, the current enrollment, the start and end time of the course and so on. |
| **Priority** | Low |
| **Dependencies** | FR2 |

## 3.2 Non-function requirements

|  |  |
| --- | --- |
| The redesign of the web pages needs to maintain the color scheme that Team Durham uses. | |
| **Security** | * After authentication, the user can only access the data within the scope of his authority and can only operate within the scope of his authority. * User / Administrator information needs to be encrypted. * Websites can withstand general malicious attacks from the Internet. Such as SQL statement injection attack, instruction guess attack. |
| **Reliability** | * The website checks the input data to prevent data anomalies. * The data will be corrected if there is a human error. * The user will be prompted if he enters invalid data. * Calendar refresh every 3 seconds. * The response time of the website is no more than 1.5 seconds. |
| **Compatibility** | * User/Administrator can use the software on any mobile devices without adjusting. |
| **Availability** | * After reading the booking system instructions, users can complete the booking within 10 minutes. |
| **Maintainability** | * Developed sites need to pass 100% coverage unit testing. * BUG modification time shall not exceed 2 working days. * No more than 200 lines of code per function. |
| **Recoverability** | * User can reset their password if they forget. * back up data each term. |
| **Robustness** | * Software can identify input error * After the abnormal situation appears, the terminal software will automatically initiate the retry mechanism, and after the abnormal situation disappears, the terminal software can recover automatically. |
| **Integrity** | * The name of the database table and the fields in the table conforms to the naming specification. * The fields in the table are complete. * The description of the fields in the database table is correct, including the type, length. * Correct relationships, indexes, primary keys, and constraints in database tables. |
| **Documentation** | * A help documentation is needed for users. * Administrator has guide documentation. * Each phrase of the project has an imprint. |

## 3.3 Functional Requirement Specification

This section specifies the use cases (*Rosenberg, 1999*) for each active user separately.



### 3.3.1 Guest Use Case

**Use case:** sign up and sign in

**Brief Description**

The guest accesses the website to sign up and sign in which allows him to have the same features as a client.

**Initial Step-By-Step Description**

Before this use case can be initiated, the Guest has already accessed the website.

1) The guest clicks on the sign-up button.

2) The guest input required information (ID, e-mail, name, telephone number, role (student or not), password) to get an account

3) The previous guest and now user can sign in with his email and password.

**Use case:** Search information

**Brief Description**

The guest can input keywords in the search text area, and it will give back all related facilities or classes/sessions and appointments.

**Initial Step-By-Step Description**

Before this use case can be initiated, the guest has already accessed the website.

1) The guest inputs keywords in the search text area.

2) The website displays the corresponding of facilities, class/sessions and appointments.

3) The guest selects the tag desired.

4) The website displays certain information.

**Use case:** Customize visualization

**Brief Description**

The guest can view calendar and select the time range for which he wants to see the facilities or classes/sessions and appointments.

**Initial Step-By-Step Description**

Before this use case can be initiated, the guest has already accessed the website.

1) The guest selects a time range(day/month/week).

2) The graph changed and information for that period is displayed.

### 3.3.2 User Use Case

**Use case:** sign in

**Brief Description**

A user can access the website and use his account credentials to sign in.

**Initial Step-By-Step Description**

Before this use case can be initiated, the user has already accessed the website.

1) The user input his ID, e-mail and password.

1) The user clicks on the sign-in button.

**Use case:** Access personal profile

**Brief Description**

After signing in, a user can go to his personal profile page where he can see his information and all the booking records, and he can also update his personal information.

**Initial Step-By-Step Description**

Before this use case can be initiated, the user has already signed in.

1) The user clicks on “personal profile” which is a link.

2) The website displays the personal profile for the user.

3) The user updates his information.

**Use case:** Search information

**Brief Description**

The user can input keywords in the search text area, and it will give back all related facilities or classes/sessions and appointments.

**Initial Step-By-Step Description**

Before this use case can be initiated, the user has already signed in.

1) The user inputs keywords in the search text area.

2) The website displays the corresponding of facilities, class/sessions and appointments.

3) The user selects the tag desired.

4) The website displays certain information.

**Use case:** Customize visualization

**Brief Description**

The user can view calendar and select the time range for which he wants to see the facilities or classes/sessions and appointments.

**Initial Step-By-Step Description**

Before this use case can be initiated, the user has already signed in.

1) The user selects a time range(day/month/week).

2) The graph changed and information for that period is displayed.

**Use case:** online booking

**Brief Description**

The users can make an online booking based on the availability of a facility.

**Initial Step-By-Step Description**

Before this use case can be initiated, the user has already signed in.

1) The user selects facilities and the available time (advice the user to book via email or telephone if the event is complex)

2) Confirmation of a booking plus the total cost is sent via email to the user.

**Use case:** forget password

**Brief Description**

If the user forgets his password, he can click on “forgot password” which is a link in login interface, and the system will send him an email which contains a link to reset password.

**Initial Step-By-Step Description**

Before this use case can be initiated, the user has already accessed the website.

1) The user clicks on “forgotten password?” which is a link in the login interface.

2)   The user needs to enter the registered e-mail and click on “reset password” button.

3)   The user will receive an instruction email which contains a link to reset password.

4)   The user enter email, new password and confirm password.

### 3.3.3 Administrator Use Case

**Use case:** Managing facilities

**Brief Description**

Administrator can add, edit, delete facilities on the website.

**Initial Step-By-Step Description**

Before this use case can be initiated, the administrator has already signed in.

1) The administrator selects facilities and the operation that needs to be performed.

2) Performing operations.

**Use case:** Updating prices of facilities

**Brief Description**

Administrator can update prices of facilities on the website.

**Initial Step-By-Step Description**

Before this use case can be initiated, the administrator has already signed in.

1) The administrator selects the facility for which the price

needs to be updated.

2) Updating the price of the facility.

**Use case:** Uploading images of facilities

**Brief Description**

Administrator can upload images of facilities on the website.

**Initial Step-By-Step Description**

Before this use case can be initiated, the administrator has already signed in.

1) The administrator selects the facility for which the image

needs to be uploaded.

2) Uploading the image of the facility.

**Use case**: Add bookings

**Brief Description**

Administrator can add a booking that is taken by email or phone.

**Initial Step-By-Step Description**

Before this use case can be initiated, the administrator has already signed in.

1) The administrator receives a booking email or phone call and collects information about the booking.

2) Adding the booking.

**Use case**: Cancel bookings

**Brief Description**

Administrator can cancel any booking manually.

**Initial Step-By-Step Description**

Before this use case can be initiated, the administrator has already signed in.

1) The administrator selects the booking that needs to be

cancelled

2) Cancelling the booking.

**Use case:** Blocking bookings

**Brief Description**

Administrator can block booking for the facilities and can also block booking of a facility in periods of time.

**Initial Step-By-Step Description**

Before this use case can be initiated, the administrator has already signed in.

1) The administrator chooses the facilities and time that need to be blocked.

2) Blocking the booking.

**Use case:** Managing announcements

**Brief Description**

Administrators can manage announcements including publish, delete, and update announcements. The administrator can sort

announcements.

**Initial Step-By-Step Description**

Before this use case can be initiated, the administrator has already signed in.

1) The administrator can add new announcements.

2) The administrator can delete and update existing announcements.

3) The administrator can sort announcements.

**Use case:** Viewing calendar

**Brief Description**

Administrator can view one overall calendar for all facilities or toggle between calendars. Administrator can view the calendar for one or more specific facilities.

Initial Step-By-Step Description

Before this use case can be initiated, the administrator has already signed in.

When the administrator opens the calendar, the calendar shows the booking status of all facilities.

1)The administrator enters specific facilities; the calendar displays the booking status for specific facilities.

### 3.3.4 Trainer Use Case

**Use case:** Booking facilities and publishing courses

**Brief Description**

Trainers can book a facility in periods of time and update details of it. The trainer must book facilities before the academic year commences.

**Initial Step-By-Step Description**

Before this use case can be initiated, the trainer has already signed in.

1) The trainer selects the facility and date of the booking and

fills in the relevant information for the course.

2) Submitting the booking form.

3) The trainer publishes the course.

4) The trainer can update the course status.

## 3.4 User Characteristics

The guest and user can see the facilities, sessions, announcements and the ranking of activities. They are also able to search particular information by typing key word.

Guest can sign up to be a user of this website.

The user can log in to the website. Once they do, they can see all bookings, activities and the available facilities in the calendar via daily, weekly and monthly views. They are also be able to make online booking for the facility.

Trainers can book facilities for a period of time and post relevant courses. In addition, trainers can update course information.

Administrator can manage bookings, facilities, events and announcements. Administrator is responsible for maintaining and editing the database as well as the web framework.

# 4. Data

## 4.1 Data Description

**Facility**

|  |  |  |
| --- | --- | --- |
| **Date item** | **Type** | **Description** |
| **F\_ID** | Int | Automatically generated unique ID for each facility |
| name | varchar | Name of each facility |
| capacity | Int | Maximum capacity of each facility |
| open | time | open time of the facility |
| close | time | close time of the facility |
| price | double | Hourly rental for the venue |
| img | string | url of facility’s image |
| info | text | information of the facility |

**Booking**

|  |  |  |
| --- | --- | --- |
| **Date item** | **Type** | **Description** |
| **B\_ID** | string | Automatically generated booking ID |
| F\_ID | Int | Facilities’ ID |
| start | datetime | start date and time |
| end | datetime | end date and time |
| usage | varchar | course / event / student booking |
| EC\_ID | int | Activity’s ID (event ID / course ID / student booking ID) |
| U\_ID | int | User’s ID |
| count | int | number of people |

**Event**

|  |  |  |
| --- | --- | --- |
| **Date item** | **Type** | **Description** |
| **E\_ID** | Int | Event ID |
| title | varchar | events title |
| url | varchar | URL of the event |
| start | datetime | event start time |
| end | datetime | event end time |
| scale | int | the scale of the event |

**Course**

|  |  |  |
| --- | --- | --- |
| **Date item** | **Type** | **Description** |
| **C\_ID** | Int | Course ID.(defaulted) |
| name | varchar | Course name |
| time | string | Weekly time of each course(e.g. 0|ss/mm/hh-ss/mm/hh|0|0|0) |
| weeks | int | Course weeks(e.g. 10) |
| start | date | Course start date |

This table is simplified course table. We assume the system could connect the university’s course table and these attributes are needed in this system.

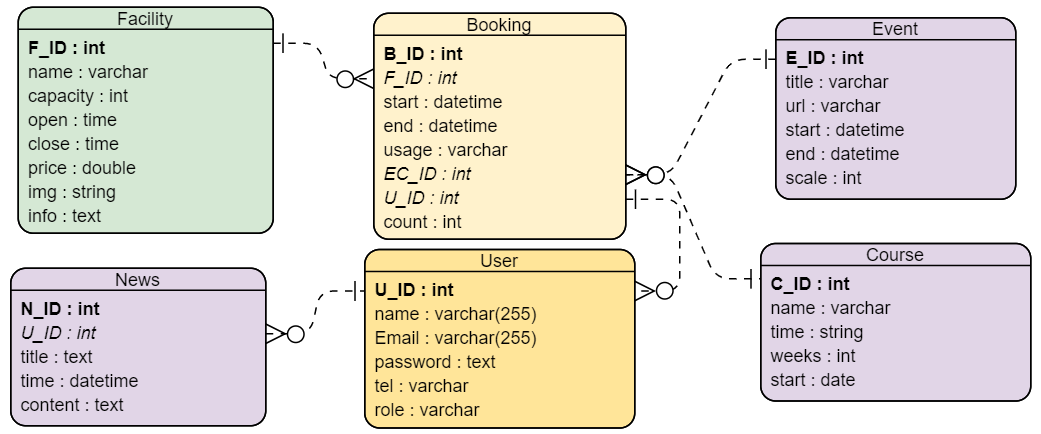
**User**

|  |  |  |
| --- | --- | --- |
| **Date item** | **Type** | **Description** |
| **U\_ID** | Int | Student ID or automatically generated user ID. (student ID starts with 0 whereas other user ID start with 1) |
| name | varchar | User’s full name |
| Email | varchar | User’s registered Email address |
| password | text | User’s encrypted password |
| tel | varchar | User’s telephone number |
| role | varchar | User’s role (student / trainer / administrator) |

**News**

|  |  |  |
| --- | --- | --- |
| **Date item** | **Type** | **Description** |
| **N\_ID** | Int | News ID |
| U\_ID | Int | Users’ ID |
| title | text | the title of the news |
| time | datetime | news published time |
| content | text | the content of the news |

## 4.2 ERD

****

# 

# 5. Definition of terms

|  |  |
| --- | --- |
| **Term** | **Description** |
| Guest | Those who are unregistered and browse only the brief content of the website. |
| Administrators | Web developers or website managers. |
| User | Website browsers with a registered account. |
| BUG | Existed errors that are waiting for developers or maintainers to fix with latest patches (Techterms.com, 2019). |
| Database | The storage of data with reasonable structure that will be used in the design and the operation of the website (Ponniah, 2010). |
| Datatype | An attribute of the data which tells the developers how they use the data in programming or database design (Hanna, 2016). |
| ERD | A diagram that shows the structure of the database and illustrates the relationships between different entity sets (Al-Masree, 2015). |
| Hashing algorithm | An approach using the reversible hash to replace the original data to encrypt information in order to protect them (Mohanty, Sarangi and Bishi, 2019). |
| Malicious attacks | A type of cyberattack using malicious software like malware, spyware, and ransomware to perform activities on a victim’s computer (Xiao et al., 2015). |
| SQL | A specific language designed for database for managing and processing data (Taylor, 2019). |

# 6. Reference

1. *Rosenberg D, Scott K. Use case driven object modeling with UML[M]. Reading: Addison-Wesley Professional, 1999.*
2. *Techterms.com. (2019). Bug Definition. [online] Available at: https://techterms.com/definition/bug [Accessed 5 May 2019].*
3. *Al-Masree, H. (2015). Extracting Entity Relationship Diagram (ERD) From Relational Database Schem. International Journal of Database Theory and Application, 8(3), pp.15-26.*
4. *Ponniah, P. (2010). Database design and development. Piscataway, NJ?: IEEE Press.*
5. *Hanna, S. (2016). An Approach to Modeling Web Services Datatype Descriptions. Journal of theoretical and applied electronic commerce research, 11(2), pp.6-6.*
6. *Mohanty, R., Sarangi, N. and Bishi, S. (2019). A secured Cryptographic Hashing Algorithm. [online] arXiv.org. Available at: http://arxiv.org/abs/1003.5787 [Accessed 7 May 2019].*
7. *Taylor, A. (2019). Sql all in one for dummies. Hoboken, N.J.: John Wiley & Sons.*
8. *Xiao, Y., Lao, S., Hou, L. and Bai, L. (2015). Mitigation of malicious attacks on network observation. International Journal of Modern Physics C, 26(10), p.1550108.*